

Report III

Miguel Rodríguez, RA: 192744, Email: m.rodriguezs1990@gmail.com

I. INTRODUCTION

El objetivo de este trabajo es estudiar el comportamiento de la tecnica PCA (Anlisis de componentes principales) para la compresion de informacion de una imagen.

El anlisis de componentes principales es una formulacion matematica, comunmente usada para la reduccion de datos [1], esta tecnica estadistica permite reducir la cantidad de datos estadisticos a analizar, para poder tomar decisiones en base solo a la informacion principal presentada por las variables. PCA es una tecnica basada en los autovalores de la matriz tratada, esta tecnica permite reducir la cantidad de valores singulares de la matriz principal para asi poder reducir la cantidad de datos a ser revisados, manteniendo sustantivamente la misma cantidad de informacion de forma reducida a traves de esta tecnica.

$$A = U\Sigma V^T = [u_1 \ u_2 \ \dots \ u_n] \begin{bmatrix} \sigma_1 & 0 & \dots & 0 \\ 0 & \sigma_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sigma_n \end{bmatrix} \begin{bmatrix} v_1^T \\ v_2^T \\ \vdots \\ v_n^T \end{bmatrix} \quad (1)$$

La obtencion de los valores singulares de la matriz se hace a traves de la descomposicion en valores singulares o *SVD* cuya descomopicion es representada por la Eq. 1, donde Σ es una matriz cuya diagonal es representada por los valores singulares ordenados de el mayor a menor, lo cual implica que mientras mayor sea el valor singular, mayor sera la informacion que este contenga de la matriz original. Esto significa, que si se eliminan los valores singulares con menor valor numerico, no sera mucha la informacion que se desprecie de la matriz original. Esta es la principal idea de la tecnica PCA, la cual dado un numero k , se eligen los k mayores valores singulares, y se vuelve a reconstruir la imagen eliminando todos los demas valores singulares, la imagen resultante sera una reconstruccion aproximada de la imagen, la cual representa la informacion mas relevante de esta.

La objetivo principal de este trabajo es aplicar la tecnica PCA en imagenes para diferentes valores de k y analizar cual es el grado de compresion y la informacion perdida en la imagen final.

II. SOLUTION

Para poder realizar este trabajo, se utilizo el algoritmo propuesto para el laboratorio, el cual puede ser visto en el Listing 1, la cual recibe una imagen RGB y el valor k para el cual se debe aplicar la tecnica PCA. Esta funcion aplica la funcion *SVD* para cada canal, y reconstruye la nueva imagen con el k ingresado. Luego de realizar esta operacion para cada canal de la imagen original, se procede a arreglar los valores de los pixeles que luego de la operacion resultaron mayores que 255 y menores que 0, cuyos valores son setted en 255 y 0 respectivamente.

```

1 def get_pca_image(img_in, k):
2     # Get dimensions of image
3     height, width, channels = img_in.shape
4
5     # Create new image
6     img_out = np.empty((height, width, channels))
7
8     # for each channel
9     for i in range(channels):
10         # Get SVD from each channel
11         U, s, V = np.linalg.svd(np.float64(img_in[:, :, i]))
12         # Create new image from each channel
13         img_out[:, :, i] = np.dot(U[:, 0:k], np.dot(np.diag(s[0:k]), V[0:k, :]))
14
15     # Normalize pixels
16     img_out[img_out > 255] = 255
17     img_out[img_out < 0] = 0
18
19     # Change image type to uint8
20     img_out = img_out.astype('uint8')
21
22 return img_out

```

Listing 1: PCA function implemented in python

Luego de reconstruir la imagen, dependiendo del k seleccionado sera de la imagen original, por lo tanto para saber cuan parecida es la imagen resultante a la imagen original, es necesario calcular el error que existe entre ambas, el cual es calculado utilizando el error medio cuadratico normalizado mostrado en la Eq. 2, el cual nos muestra cuan parecida es una matriz a otra en valores entre el $[0, 1]$, lo cual nos permite transportar estos valores a valores porcentuales.

$$NMSE(I_1, I_2) = \sum \frac{(I_1 - I_2)^2}{I_1^2} \quad (2)$$

En paralelo con el calculo del error, tambien es necesario saber cuanta es la compresion realizada por la tecnica, por lo cual para esto calcularemos una media normalizada expresada en la Eq. 3, la cual calcula la cantidad de memoria en porcentaje que esta siendo utilizada con respecto a la cantidad original.

$$p = \frac{\text{Memory required to represent } I_{pca}}{\text{Memory required to represent } I_{\text{original}}} \quad (3)$$

Para el calculo de la cantidad de memoria utilizamos dos metricas, la primera es la cantidad de memoria ram utilizada por la imagen original luego de ser sometida a la tecnica *SVD*, y la cantidad de memoria ram de la nueva imagen luego de descartar todos los valores singulares. La segunda metrica utilizada es el peso de las imagenes al ser guardadas y codificadas por algun metodo de compresion de imagenes (*PNG*, *JPG*, *etc.*). Ambas metricas son calculadas por medio de la Eq. 3.

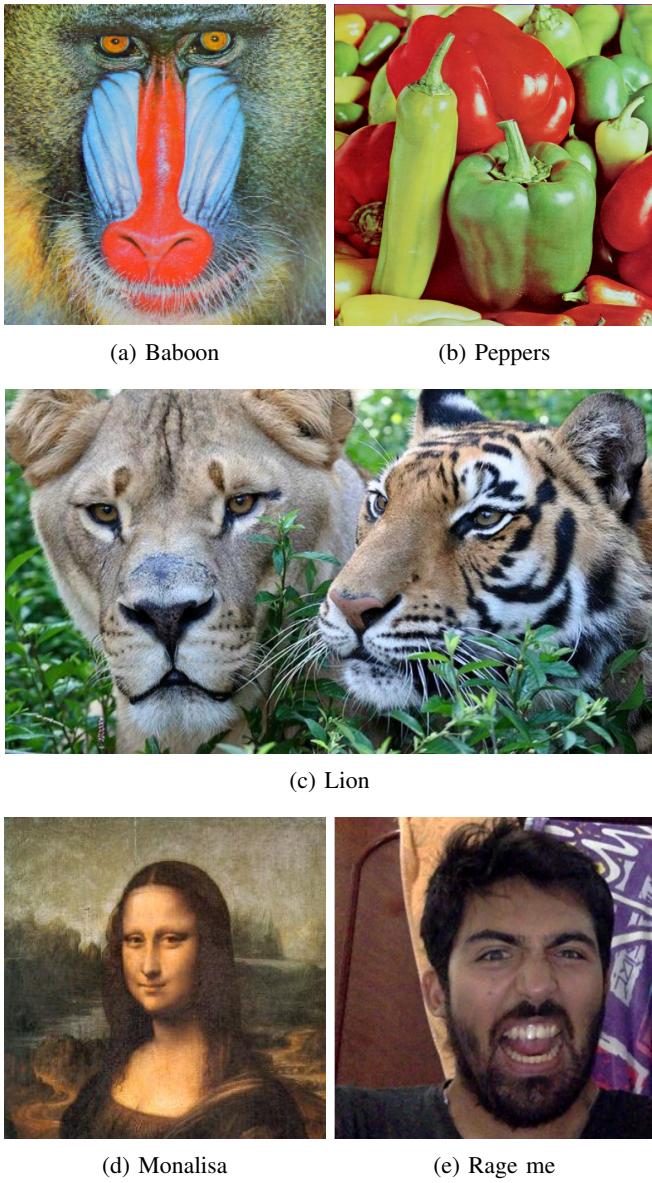


Fig. 1: Original images used into experiments.

Para poder realizar una experimentacion mas profunda sobre el comportamiento de este metodo sobre las imagenes, decidimos realizar el calculo de estas metricas, para cada uno de los posibles valores que se peuden tomar de k en las imagenes a experimentar.

III. EXPERIMENTS

En esta seccion mostraremos los experimentos realizados en este trabajo, para ver el desempeo de la tecnica PCA, para esto utilizamos cinco imagenes distintas, con distintos tamaos y distintos formatos (*PNG* y *JPG*), las cuales son mostradas en la Fig. 1. Para la Fig. 1a, se realizaron dos experimentos,

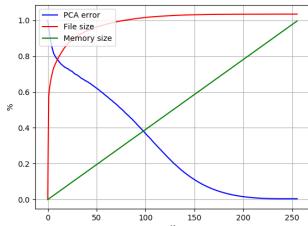
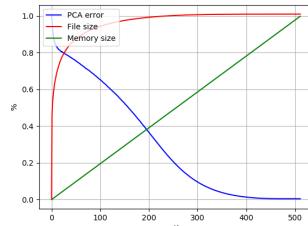
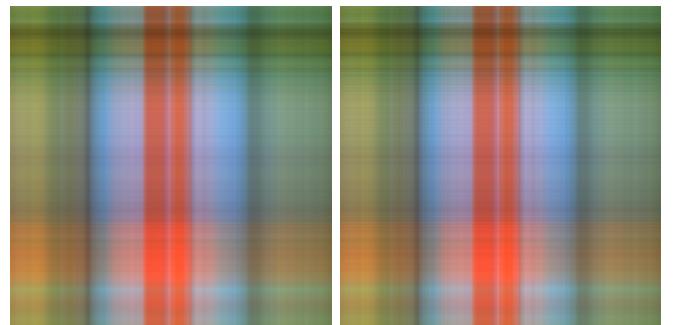
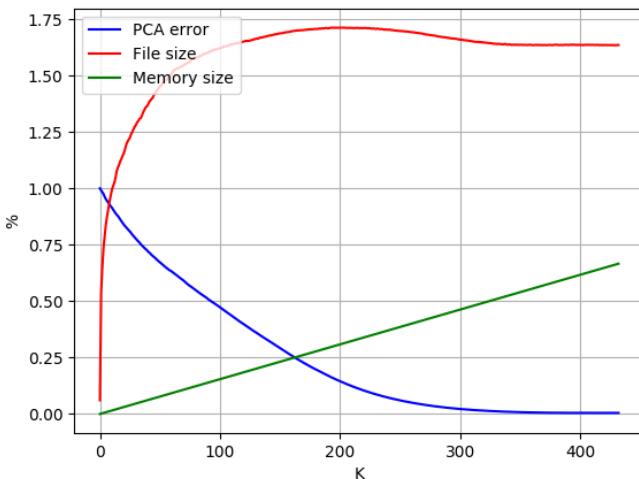
una donde la imagen es de tamao 256×256 y otra de tamao 512×512 , asi mismo a la Fig. 1e, se utilizo una version de la imagen en *PNG* y otra *JPG*.

Los graficos desplegados en la Fig. 2, muestran como es la relacion entra la Eq. 2 y las dos aplicaciones de la Eq. 3, para cada uno de los posibles k para la imagen. En las graficas se puede ver que el error es inversamente proporcional a la relacion de los tamaos a medida que el valor de k va aumentando. Tambien se puede obsevar que existen dos puntos de intersección en los errores.

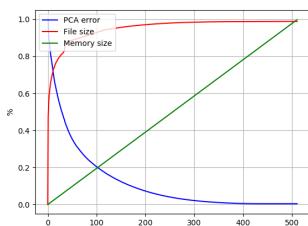
El primer experimento realizado se utilizo la Fig. 1a, en dos versiones de distintos tamaos 256×256 y 512×512 ambas en formato *PNG*. Como se puede observar en la Fig. 3, mientras mayor sea el numero de k , mayor informacion contendra la imagen resultante del proceso PCA. En los graficos mostrados en las Figs. 2a and 2b, se puede ver que existen dos puntos de intersección para la curva de error, la primera es la intersección con la curva de relacion de temao del archivo, la cual esta representada en $k = 10$ y $k = 26$ respectivamente, si se ven las Figs. 3c and 3d, se puede observar que imagen resultante para esos valores de k contiene bastante informacion visual, por lo cual se podria decir que ya es una imagen aceptable. La segunda intersección de la curva de error es con la curva de la relacion de la memoria utilizada, la cual es en $k = 98$ and $k = 196$ respectivamente, cuyas imagenes estan representadas en las Figs. 3e and 3f, las cuales al ser analizadas casi no tienen diferencia visual con respecto a la imagen original, siendo el error menor al 40% la imagen las imagenes resultantes tienen resultados bastante aceptables. En los graficos tambien se puede observar que la relacion de los tamaos de los archivos luego de un cierto k sobrepasa el 100%, por lo cual se puede decir que para esos k , el archivo de salida PCA, tiene un mayor tamao en bytes que el archivo original, esto debido a la codificacion utilizada para guardarlo.

El segundo experimento se realizo utilizando una imagen en formato *JPG*, la cual puede ser observada en la Fig. 1c. El grafico resultante de este experimento puede ser visualizado en la Fig. 2c, el cual tiene un comportamiento muy particular, tanto en la relacion existente en los archivos de salida como en la cantidad de memoria utilizada. Como se puede visualizar en la linea roja (la cual representa la relacion de los archivos de salida) logra tomar valores cercanos al 175% para algunos k , luego esto baja y se normaliza en valores cercanos al 160%, con lo cual se puede observar que la codificacion de la imagen resultante al proceso PCA es un caso donde el proceso de codificacion *JPG* se comporta peor que con la imagen original. Por otro lado la linea verde (la cual representa la relacion de la cantidad de memoria utilizada por las imagenes), esta no tiene el mismo comportamiento que los otros experimentos, esto debido que al llegar al ultimo valor posible de k , el valor de la relacion no se aproxima a uno, sino que se aproxima a 0.7, con lo cual se podria decir que la cantidad de memoria que se utiliza para representar esta imagen despues del proceso de PCA es menor que la cantidad de memoria utilizada por la imagen original.

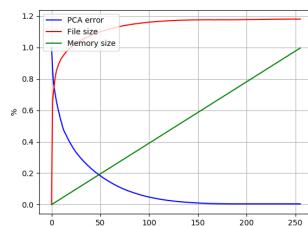
El tercer experimento fue realizado utilizando la Fig. 1b, la cual esta en formato *PNG*. El grafico resultante de este experimento puede ser observado en la Fig. 2d, en la cual

(a) Graphic of Baboon 256×256 (b) Graphic of Baboon 512×512 (a) Baboon 256×256 , $k = 1$ (b) Baboon 512×512 , $k = 1$ 

(c) Graphic of Lion



(d) Graphic of Peppers



(e) Graphic of Monalisa

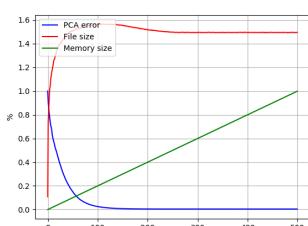
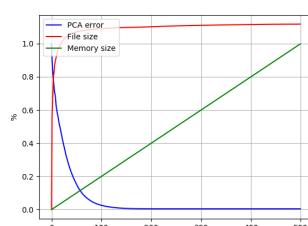
(f) Graphic of rage me in *JPG*(g) Graphic of rage me in *PNG*

Fig. 2: PCA graphics. La linea azul representa el error propuesto por la Eq. 2, la linea roja es razon de bytes utilizados en los archivos de la imagen original y imagen final y la linea verde representa la razon de bytes utilizados en memoria por la representacion SVD de la imagen original y la imagen final.

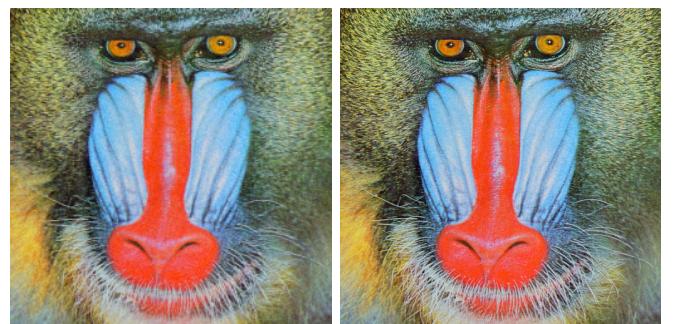
(c) Baboon 256×256 , $k = 10$ (d) Baboon 512×512 , $k = 26$ (e) Baboon 256×256 , $k = 98$ (f) Baboon 512×512 , $k = 196$

Fig. 3: PCA baboon's images. A la izquierda los resultados de aplicar PCA en la Fig. 1a de tamano 256×256 y a la derecha los resultados de aplicar PCA en la Fig. 1a de tamao 512×512

se puede ver el comportamiento ideal y esperado para esta tecnica, donde la relacion de los tamaos de los archivos al llegar al k maximo, es igual a 1 y la relacion de la memoria utilizada tambien. En la Fig. 5, se puede observar las imagenes resultantes para distintos valores de k , los cuales con $k = 1$ (See Fig. 5a) no se puede observar nada de informacion util, con $k = 11$ (See Fig. 5b) se puede observar ya la forma general del contenido de la imagen, este valor de k , es donde se intersectan las curvas de error y la relacion entre los archivos. Para $k = 103$ (See Fig. 5d) es donde se intersectan las curvas de error y la relacion entre la memoria utilizada, la imagen que se ve es muy nitida y parece mucho a la original, pero

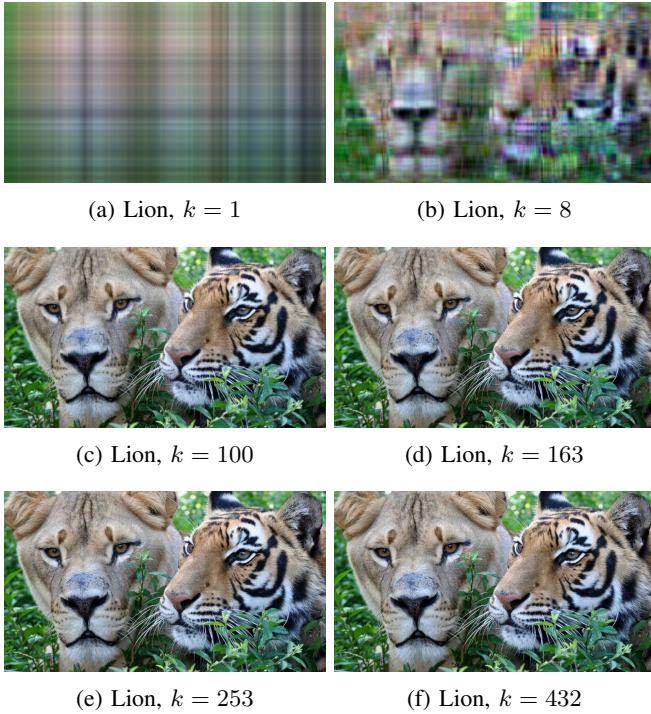


Fig. 4: PCA Lion's images. La Fig. 4b, con $k = 8$, es donde se produce la intersección entre el error con la relación de los archivos. La Fig 4d con $k = 163$ es la resultante de la intersección del error con la relación del tamaño utilizado en la memoria.

para valores menores de k tambien se puede lograr una nitidez aceptable, eg. en la Fig. 5c, la cual muestra una imagen muy aceptable que puede ser comparada con la original y no tener mucha diferencia.

El cuarto experimento fue realizado en una imagen de formato *PNG*, la cual puede ser visualizada en la Fig. 1d. El grafico resultante de este experimento puede ser observado en la Fig. 2e, en el cual se puede ver que el comportamiento de esta imagen es muy parecido al experimento anterior, pero la curva de la relación de los tamaños de los archivos después de un cierto k , se vuelve mayor que uno, por lo cual la codificación *PNG* hace que la imagen al ser guardada tenga un mayor tamaño que la imagen original, el comportamiento de las demás curvas es el esperado. En las Fig. 6 se puede ver las salidas para distintos k , siendo $k = 3$ y $k = 49$ (See Figs. 6b and 6d)) las intersecciones de las curvas de error con las relaciones de los archivos y la memoria utilizada respectivamente.

En el quinto y ultimo experimento se realizó utilizando la imagen mostrada en la Fig. 1e. Para este experimento se realizó con la misma imagen en dos formatos distintos (*PNG* and *JPG*). Los resultados de estos experimentos pueden ser vistos en los graficos desplegados en las Figs. 2f and 2g, en los cuales se puede observar que las curvas tienen un com-

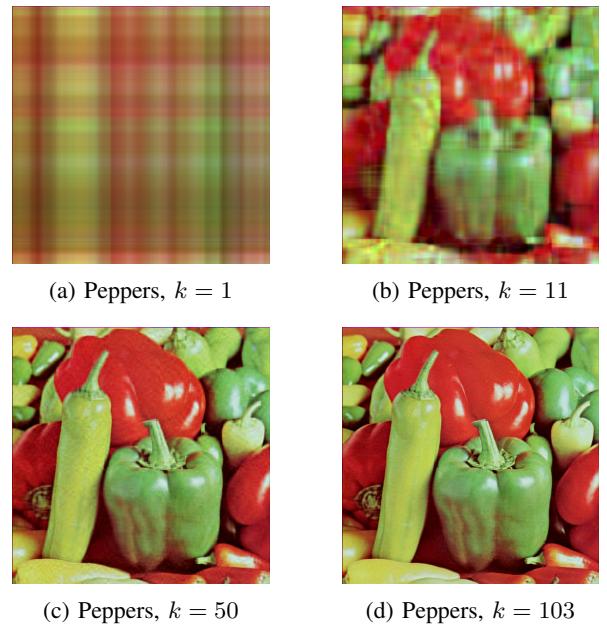


Fig. 5: PCA Peppers's images. La Fig. 5b, con $k = 11$, es donde se produce la intersección entre el error con la relación de los archivos. La Fig 5d con $k = 103$ es la resultante de la intersección del error con la relación del tamaño utilizado en la memoria.

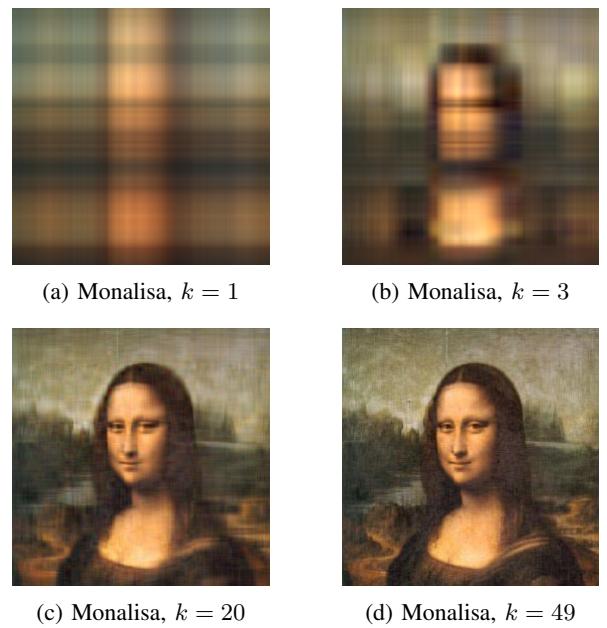


Fig. 6: PCA Mona Lisa's images. La Fig. 6b, con $k = 3$, es donde se produce la intersección entre el error con la relación de los archivos. La Fig 6d con $k = 49$ es la resultante de la intersección del error con la relación del tamaño utilizado en la memoria.

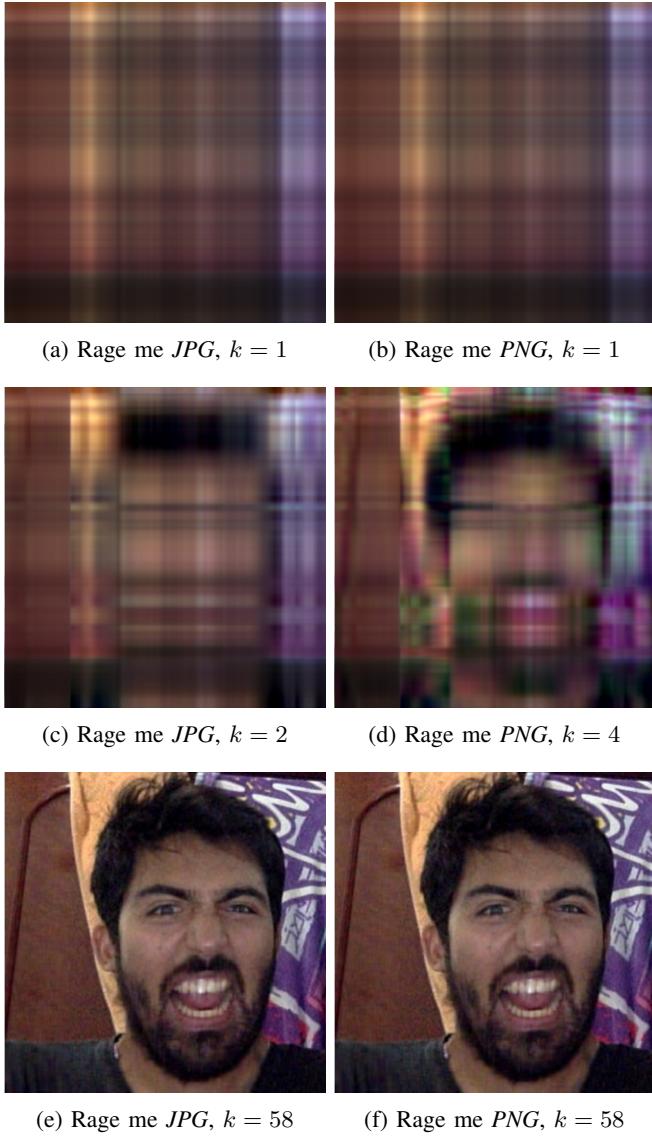


Fig. 7: PCA Rage me's images. A la izquierda los resultados de aplicar PCA en la Fig. 1e con formato *JPG* y a la derecha los resultados de aplicar PCA en la Fig. 1e con formato *PNG*.

portamiento muy parecido al mostrado en los experimentos anteriores, donde la curva de relacion entre los archivos de salida para ciertos k supera el valor uno, por lo cual el archivo nuevo es de mayor peso que el original. Al igual que en el tercer experimento, el cual tambien fue realizado con una imagen en formato *JPG*, la curva de relacion de los tamaños de los archivos, experimento un crecimiento y posteriormente un decrecimiento, el cual al parecer es caracteristico de las figuras en este formato. En la Fig. 7 se puede ver el resultado de aplicar la tecnica PCA para distintos valores de k en ambos formatos. La interseccion entre las curvas de error y la relacion entre el tamaos de los archivos puede ser visualizada en $k = 2$

en la imagen con formato *JPG* (See Fig. 7c) y $k = 4$ en *PNG* (See Fig. 7d), donde se puede observar en las imagenes que algo de informacion de la imagen original puede ser vizualizada. Asi mismo, en ambos formatos para $k = 58$ se produce la intersección entre las curvas de error y la curva de relacion de la memoria utilizada, donde se puede visualizar que la imagen obtenida tiene la mayor parte de la informacion relevante de la imagen original.

IV. CONCLUSION

La tecnica PCA es una tecnica matematica poderosa para poder realizar una reduccion de la informacion que se tiene sobre ceirto fenomeno, en el caso de las imagenes, al aplicar esta tecnica en la compresion de datos como se observo en los experimentos realizados (See Section III), se puede concluir que no es una tecnica completa, esto debido a que se debe mejorar para poder obtener una compresion mejor.

Con respecto al tamao de las imagenes, la tecnica tiene un comportamiento muy parecido para imagenes que muestran la misma informacion visual, pero tienen distintos tamaos. Asi puede ser visto en el primer experimento (See Fig. 2a and 2b), donde los graficos resultantes de la aplicacion de la tecnica a ambas imagenes resultaron ser muy parecidos.

La tecnica tiene un comportamiento muy estable con imagenes que son guardadas en archivos en formato *PNG*, no asi con las imagenes que son guardadas en formato *JPG*, esto debido a que la relacion de los tamaos de los archivos tiene variaciones muy intestables para este formato.

A su vez si se analiza el comportamiento de la tecnica para una misma imagen pero guardada en formatos distintos (*PNG* and *JPG*), se puede observar que el comportamiento de la compresion es muy parecido, pero que el formato *JPG* tiene distintas variaciones intestables.

Con respecto a la elección del valor de k , creemos que la elección de este valor debe ser un valor que tenga un error menor al 75%, y que este entre el conjunto de valores que estan entre la intersección del error con ambas relaciones, esto debido a que como se vio en todos los experimentos, la imagen resultante de la intersección la relacion de tamaos de los archivos con el error, es una imagen que muestra las principales características de la imagen original, pero no se logra ver visualmente identica con respecto a la original. Por el contrario la imagen resultante de la intersección entre el error y la relacion de la utilizacion de memoria, es una imagen muy nitida y muy parecida a la original, por lo tanto proponemos que la elección del valor k debe estar entre estos dos dependiendo del grado de error que se requiera en la imagen.

REFERENCES

- [1] I. Jolliffe, *Principal Component Analysis*. Springer Verlag, 1986.